NETCONF: THE PROGRAMMABLE INTERFACE FOR SDN AND NFV

tail-f

Table of Contents

Executive Summary

Network operators can clearly envision the Software Defined Networking (SDN) services model of the future: All network elements are part of a single, programmable fabric. Services can be assembled from a wide range of virtualized multivendor devices on the fly. They can be designed at a high level, independent of the complexities and device dependencies of the heterogeneous underlying infrastructure. Services can be automated end-to-end and provisioned in minutes.

This model is now achievable through the power of SDN and Network Function Virtualization (NFV). Implementing it, however, requires a fully programmable network. Virtualized Network Functions (VNFs) must be remotely programmable by a SDN controller, and configurations and changes must be entirely automated, without requiring manual intervention by human network operators. To make this possible, VNF vendors across the industry must evolve their virtualized devices to be programmable in SDN environments. To do it, they need a standard configuration management protocol that can provide secured, reliable transport and support network-wide transactions in the creation, modification, and deletion of configuration information in VNFs.

Now, the industry has zeroed in on NETCONF, along with the YANG data modeling language, to provide this standard configuration protocol. NETCONF overcomes the shortcomings of alternative protocols for device automation. Originally designed as a management protocol, it provides a simple, standardized way to enable a programmable interface to any device.

A wide range of SDN projects and industry organizations, such as the Open Networking Foundation (ONF), the Open Daylight Project, the Open Network Operating System (ONOS) Project, and the Open Platform for NFV (OPNFV) are now promoting NETCONF as a universal southbound interface for the configuration and management of both VNFs and physical network devices in SDN environments. Now, it's time for device and VNF vendors to take the next step. By adding NETCONF support to your VNF, it becomes possible to deploy it in any standards-based SDN environment. Your customers will be able to take advantage of your NFV innovations as part of their programmable, fully automated networks.

What's Wrong with Other Southbound Intertaces?

In the past, network operators used two primary strategies to configure southbound devices: SNMP and device-specific Command Line Interface (CLI). Neither is suitable for the automated, programmable networks that operators are now trying to implement.

Long before SDN, SNMP was widely deployed in networks, mostly for device monitoring. However, as described in RFC 3535, SNMP's stateless nature and its inability to support a robust transactional model and roll-back of transactions limits its applicability to modern programmable networks. Additionally, with SNMP configuration, there is no defined discovery process for finding the correct MIB modules supported by the network devices. As a result of these shortcomings, SNMP writes are typically not being defined in industry-standard MIBs. If writable objects are defined, network operators are not using them.

The other traditional alternative for automating configuration changes to the network has been CLI scripting. Most, if not all, network devices typically support their own flavor of CLI. However, CLI scripting has several limitations including the lack of transaction management and structured error management. Additionally, the ever-changing structure and syntax of commands make scripts fragile and costly to maintain. Ultimately, CLI scripts were designed to be used manually by human network operators, not as an API for programmatic access.

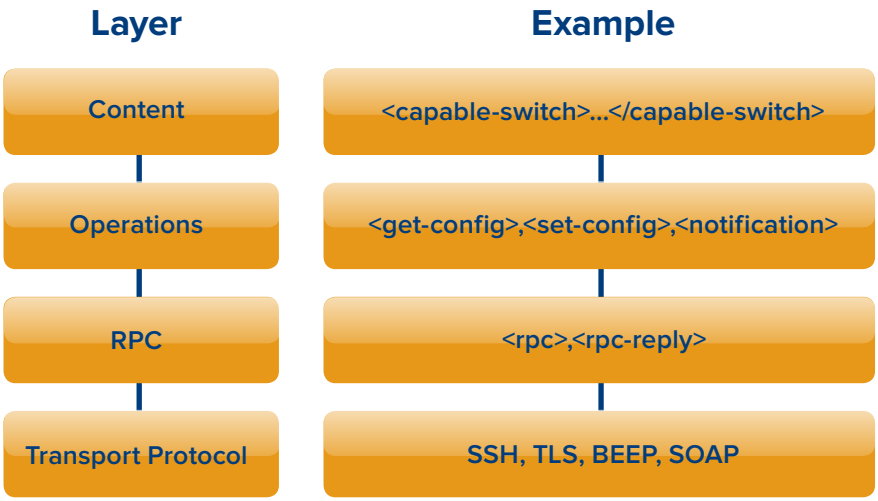
In more recent years, the RESTful API over HTTP has been used for configuration management, using HATEOAS (Hypermedia As The Engine Of Application State) principles. However, REST does not use any data modeling language to define the application-specific content of the API. This causes significant API performance inefficiencies. Also, REST is restricted to syntactic modeling. Without the schema-level semantics and validation constraints that are available via YANG, operators can't formally model (or effectively automate) end-to-end services.

Currently, the IETF is working to define a more viable REST alternative called RESTCONF (1), which is currently in its draft form and will soon be a RFC. RESTCONF provides a framework for standards-based mapping from YANG data models to the world of REST and HTTP. As a simplified alternative to NETCONF, it provides a programmatic Web-based interface to access data defined in YANG — well-suited to Web-based applications that follow REST principles and are compatible with resource-oriented device abstraction. As a result, RESTCONF is expected to find wide use in the SDN community as a northbound interface for SDN controllers. However, RESTCONF is not intended to replace NETCONF for southbound device configurations.

Due to REST's stateless nature, RESTCONF won't be as robust as NETCONF and will only provide a comparatively simplified transactional model. The methods as defined by RESTCONF are used to edit data resources, typically only one data resource instance at a time, represented by YANG data models. This represents a limited subset of the transactional capabilities of the NETCONF protocol. For these reasons, most industry organizations now agree that, while RESTCONF can provide an effective interface to northbound management and OSS systems, NETCONF provides the best solution for SDN-enabling southbound devices.

What is NETCONF

NETCONF (NETwork CONFiguration Protocol) 1.1 is defined in RFC 6241 (2). It uses Extensible Markup Language (XML)-based data encoding for the configuration data as well as protocol messages. NETCONF protocol operations are realized as remote procedure calls (RPCs). As a result, NETCONF can provide a robust programmable interface into the network element. The diagram below shows the various layers of the NETCONF protocol.



NETCONF was designed to enable:

- Distinction between configuration and state data
- Multiple configuration data stores (candidate, running, startup)
- Configuration change transactions
- Configuration testing and validation support
- Selective data retrieval with filtering
- Streaming and playback of event notifications
- Extensible remote procedure calls

Basic NETCONF operations include the following:

- Get configuration <get-config>
  Retrieve all or part of a specified configuration from a named data store.
- Get all information <get>
  Retrieve running configuration and device state information.

- Edit configuration <edit-config>
  Load all or part of a specified configuration to the specified target configuration.

- Copy configuration <copy-config>
  Create or replace an entire configuration data store with the contents of another complete configuration data store.

- Delete configuration <delete-config>
  Delete a configuration data store

- Lock and unlock <lock>, <unlock>
  Activate a short-lived lock and unlock of the configuration system of a device.

- Close and kill session <close-session>, <kill-session>
  Gracefully (close) or force (kill) termination of a NETCONF session.

NETCONF's companion data modeling language, YANG, is defined in RFC 7950 (3).

Unlike other configuration protocols, NETCONF provides a confirmed commit capability for network-wide transactions. It treats configuration changes across multiple VNFs as a single transaction, ensuring that either all configuration changes are applied or none are. If a set of configuration changes being applied to multiple VNFs results in loss of network connectivity, preventing the NETCONF client from confirming the commit action, the set of NETCONF-enabled VNFs will automatically roll back their unconfirmed configuration changes without requiring a truck roll. This capability alone promises to dramatically reduce errors, operational costs, and service delivery timelines for network operations.

Additionally, with NETCONF the network management/OSS layer is no longer responsible for error handling and recovery. Historically, this has been a major source of operational costs and delays for network operators, as OSS teams had to develop extensive device-specific adapters and CLI scripts. With NETCONF, error handling is now shifted down to the VNFs or physical network devices themselves. Central management/OSS layers can focus on service orchestration, network-wide transactions, and recovery at the network level. This significantly simplifies the configuration management capabilities needed in a SDN controller (or the EMS/NMS/OSS) and reduces development costs.

Another huge advantage of NETCONF's transaction-oriented approach is that configuration changes are made independent of their order provided during transaction preparation. This is a major departure from the traditional CLI approach, where inputting changes in the wrong order causes configurations to fail. Indeed, this requirement to input changes in a specific order is one of the main reasons that CLI scripting is such a daunting task. It places unnecessary rules on operators (for example, requiring that an interface be enabled first before its protocol can be enabled), and is a major cause of errors and delays when turning up or changing network services. By enabling order-independent configuration changes, NETCONF provides huge cost savings for operators automating their networks.

NETCONF Is Now Widely Adopted by the SDN Community

As the creators of the SDN architecture (4) point out themselves, SDN isn't so much about the invention of something new. Rather, it is about the simplification and abstraction of the entire network into three networking layers: Specification, Network Operating System, and Forwarding Layers. This architecture hides the complexities of the distributed control protocols from being exposed to the OSS layer—an essential requirement for automating and simplifying networks.

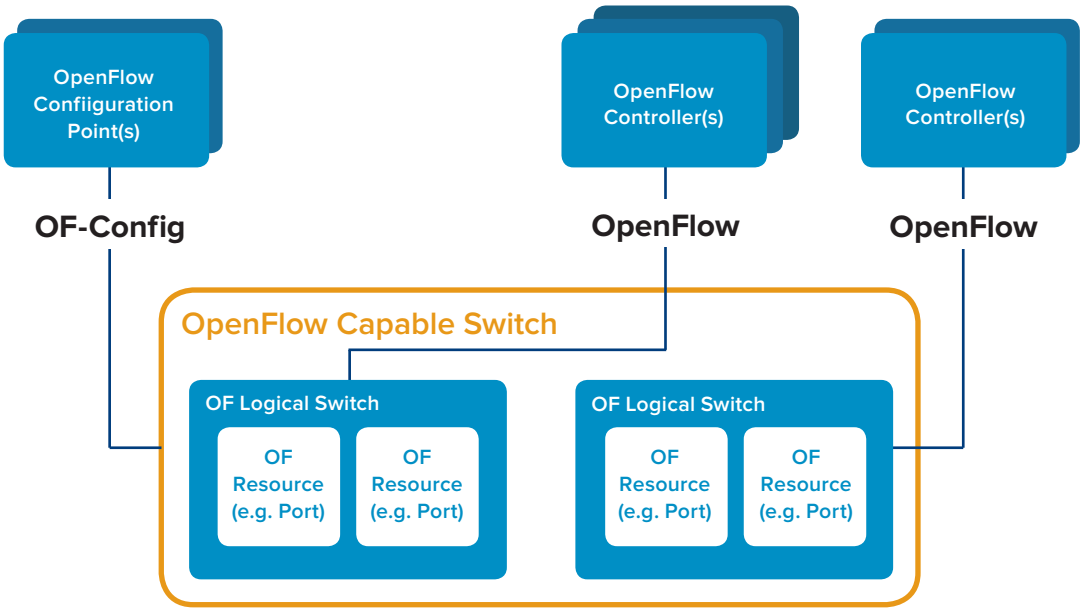
The Specification Layer provides details to describe the networking needs of the end-customer. The Network Operation System Layer then translates these into concrete device settings to be sent down into the individual network elements. For the Forwarding Layer, OpenFlow is merely one of many choices in addition to all existing forwarding-layer protocols including the IP layer.

The major organizations in the SDN community have now selected NETCONF as the network management protocol of choice to make networking devices or VNFs programmable and to simplify OSS automation. The following sections provide details.

Open Network Foundation

ONF is a user-driven organization dedicated to the promotion and adoption of SDN through open standards development. In addition to introducing the OpenFlow Standard, which enables remote programming of the forwarding plane, ONF also defined the OpenFlow Management and Configuration Protocol (OF-CONFIG 1.2 (5)) specification as its companion protocol to enable the remote configuration of OpenFlow-capable Switches. In OF-CONFIG 1.2, ONF requires that any devices supporting OF-CONFIG must implement the NETCONF protocol as their transport protocol.

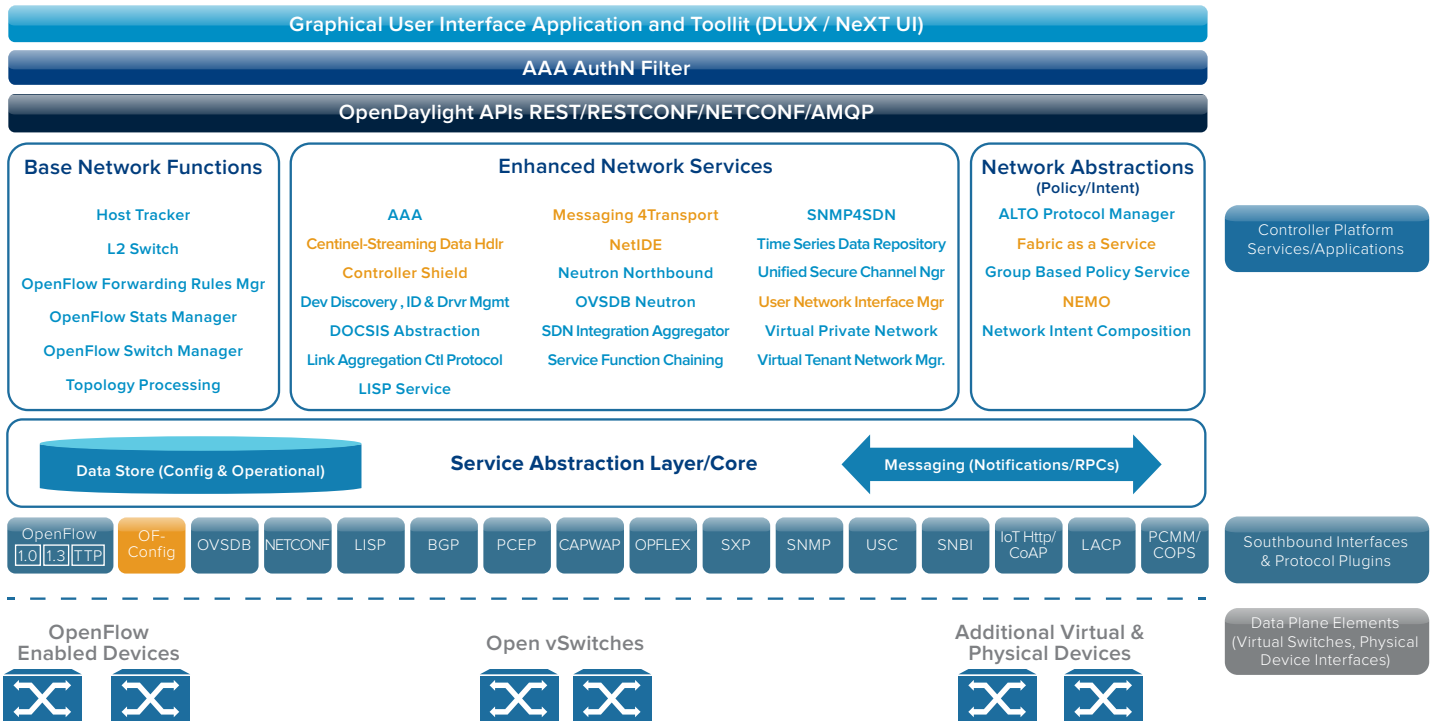
The following diagram illustrates the relationship between components defined in OF-CONFIG 1.2, the OF-CONFIG protocol, and the OpenFlow protocol.



OpenDaylight

The OpenDaylight Project is a collaborative open-source project hosted by the Linux Foundation. The goal of the project is to accelerate the adoption of SDN and create a solid foundation for NFV. This project establishes an open and reference framework for programmability and control through an open-source SDN and NFV solution. As part of the OpenDaylight Controller project, NETCONF is being used as a southbound interface to remotely configure NETCONF-enabled VNFs (6). It is expected that the latest release of ODL, Beryllium, will be embedded in more than 20 commercial products and solutions. It will also be used in the Open Platform for NFV (OPNFV), which is a carrier-grade, integrated, open-source platform to accelerate the introduction of new NFV products and services.

The following diagram illustrates how NETCONF sits in the Southbound Interfaces and Protocol Plugins layer of the OpenDaylight architecture in the Beryllium Release.



Open Network Operating System Project

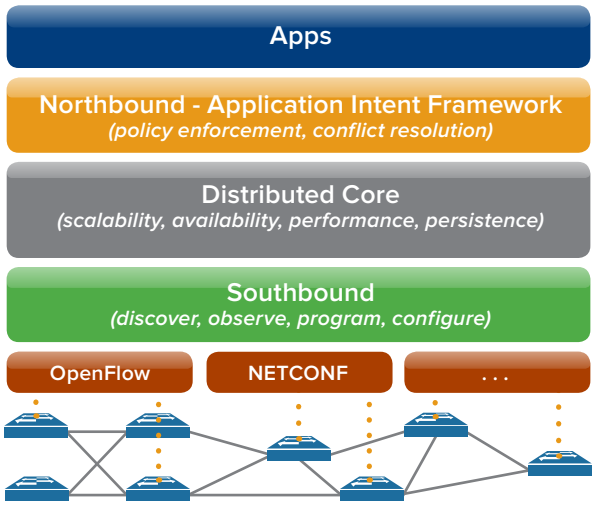
The ONOS, with engineering resources provided by ON.Lab, is the first open-source SDN network operating system targeted specifically at service provider and mission-critical networks. ONOS is purpose-built to provide the high availability (HA), scale-out, and performance that these networks demand. In addition, ONOS has created useful northbound abstractions and APIs to simplify application development, and southbound abstractions and interfaces for control of OpenFlow-ready and legacy devices. As part of the southbound interface architecture of ONOS, NETCONF is supported by its Southbound Interface Layer (7).

The following diagram illustrates where NETCONF sits in ONOS' layered architecture.

Northbound Abstraction:
- network graph
- application intents

Core:
- distributed
- protocol independent

Southbound Abstraction:
- generalized OpenFlow
- pluggable & extensible



At the time of this writing, there has been a formal announcement made on the merger of ON.Lab and ONF, promising to combine the best of both projects' standards and open-source processes to further the development of SDN.

Conclusion

Network operators worldwide have embraced the vision of SDN network programmability. By replacing device-specific CLI scripting with a programmable API into every multivendor physical and virtual device in their networks, they can automate their environments, accelerate service timelines, and lower costs and errors. However, network operators can't accomplish this on their own. They need the assistance of VNF vendors to make their virtualized devices programmable.

NETCONF provides the most flexible and effective mechanism to accomplish this. NETCONF, along with the YANG modeling language, allows for formal semantic modeling of end-to-end services. NETCONF's transactional configuration model and confirmed commit capability enable the automation of network-wide transactions, reducing errors and operational complexity. For these reasons, service providers, industry groups, and open-source SDN projects have selected NETCONF as the programmable interface of choice between SDN controllers and southbound devices. Indeed, a growing number of Tier-1 service providers now require that any new devices deployed in their networks be NETCONF-enabled.

VNF and physical device vendors worldwide have heard the SDN and service provider community and are taking action. If you have not already NETCONF-enabled your VNFs, it's time to do so. By adding NETCONF support to your physical and virtual devices, you help your customers realize significant operational savings. You make it possible for your products to be deployed as part of fully automated, agile service provider networks. You make it much easier for service providers and their customers to capitalize on your NFV innovations.

For More Information

Tail-f Systems offers the broadest family of NETCONF-enabled products and tools in the industry. Our industry-leading ConfD Management Agent is used by VNF and device vendors worldwide to make their products SDN-programmable. With our YANG data model-driven solution, you can quickly and inexpensively deliver NETCONF support in your products, along with other automatically rendered northbound interfaces.

Tail-f Systems' implementations of NETCONF are mature and field-proven in Tier-1 service provider production networks. Our participation in and contribution to the development of the NETCONF and YANG standards ensures that you and your customers will benefit from a fully standards-compliant, future-proof solution.

For more information about ConfD, visit http://www.tail-f.com/management-agent/.

References

1. RESTCONF Protocol, draft-ietf-netconf-restconf-17, September 28, 2106.

2. Network Configuration Protocol (NETCONF), RFC 6241, June 2011.

3. The YANG 1.1 Data Modeling Language, RFC 7950, August 2016.

4. SDN Abstractions, http://www.slideshare.net/martin_casado/sdn-abstractions, June 5, 2011.

5. OpenFlow Management and Configuration Protocol (OF-CONFIG) 1.2, Open Networking Foundation, 2014.

6. OpenDaylight's Cloud and NFV Use Cases, https://www.opendaylight.org/cloud-and-nfv, October 26, 2016.

7. ONOS Southbound: Protocol, Providers, Drivers, https://wiki.onosproject.org/display/ONOS/Southbound%3A+Protocol,+Providers,+Drivers, April 29, 2016.

tail-f

www.tail-f.com
info@tail-f.com

Corporate Headquarters

Sveavagen 25
111 34 Stockholm
Sweden
+46 8 21 37 40